

Geo

The purpose of this module is to provide a functionality that is quite similar to the `map` directive – affecting a variable based on the client data (in this case, the IP address). The syntax is slightly different in that you are allowed to specify IPv4 and IPv6 address ranges (in CIDR format):

```
geo $variable {
    default unknown;
    127.0.0.1    local;
    123.12.3.0/24 uk;
    92.43.0.0/16 fr;
}
```

Note that the preceding block is being presented to you just for the sake of the example and does not actually detect U.K. and French visitors; you'll have to use the GeoIP module if you wish to achieve proper geographical location detection. In this block, you may insert a number of directives that are specific to this module:

- `delete`: Allows you to remove the specified subnetwork from the mapping.
- `default`: The default value given to `$variable` in case the user's IP address does not match any of the specified IP ranges.
- `include`: Allows you to include an external file.
- `proxy`: Defines a subnet of trusted addresses. If the user IP address is among the trusted ones, the value of the `X-Forwarded-For` header is used as an IP address instead of the socket IP address.
- `proxy_recursive`: If enabled, this will look for the value of the `X-Forwarded-For` header even if the client IP address is not trusted.
- `ranges`: If you insert this directive as the first line of your `geo` block, it allows you to specify IP ranges instead of CIDR masks. The following syntax is thus permitted: `127.0.0.1-127.0.0.255 LOCAL;`

GeoIP

Although the name suggests some similarities with the previous one, this optional module provides accurate geographical information about your visitors by making use of the *MaxMind* (<http://www.maxmind.com>) GeoIP binary databases. You need to download the database files from the MaxMind website and place them in your Nginx directory.



This module is not included in the default Nginx build.

All you have to do then is specify the database path with one of the following directives:

```
geoip_country country.dat; # country information db
geoip_city city.dat; # city information db
geoip_org geiorg.dat; # ISP/organization db
```

The first directive enables several variables: `$geoip_country_code` (two-letter country code), `$geoip_country_code3` (three-letter country code), and `$geoip_country_name` (full country name). The second directive includes the same variables, but provides additional information: `$geoip_region`, `$geoip_city`, `$geoip_postal_code`, `$geoip_city_continent_code`, `$geoip_latitude`, `$geoip_longitude`, `$geoip_dma_code`, `$geoip_area_code`, and `$geoip_region_name`. The third directive offers information about the organization or ISP that owns the specified IP address by filling up the `$geoip_org` variable.



If you need the variables to be encoded in UTF-8, simply add the `utf8` keyword at the end of the `geoip_` directives.

UserID filter

This module assigns an identifier to the clients by issuing cookies. The identifier can be accessed from the variables `$uid_got` and `$uid_set` further in the configuration.

Directive	Description
userid Context: http, server, location	Enables or disables issuing and logging of cookies. The directive accepts four possible values: <ul style="list-style-type: none"> • <code>on</code>: Enables v2 cookies and logs them • <code>v1</code>: Enables v1 cookies and logs them • <code>log</code>: Does not send cookie data, but logs the incoming cookies • <code>off</code>: Does not send cookie data Default value: <code>userid off;</code>